Principle Investigator Name: Carl Hewitt
PI Institution: MIT
PI Phone Number: 617-253-5873
PI E-mail address: hewitt@ai.mit.edu
Grant or Contract Title: The Actor Model
Grant or Contract Number: N00014-88-K-0460
Reporting Period: Final Report

1 Summary of Technical Progress

Objectives

The objective of the Actor Model project is to contribute to the science of large scale concurrent systems and the technology which is enabled by this science. This work is based on the Actor Model, a universal paradigm for concurrent computation in which primitive computational entities (actors) communicate by passing messages. The inherent concurrency, reconfigurability, and asynchronous nature of the actor paradigm make it ideally suited for modeling and constructing sophisticated large scale concurrent systems on general purpose concurrent computers. This actor paradigm can also be used as a foundation for studying and implementing other diverse concurrency paradigms. The Actor Model project involves study of the Actor model, implementation of a prototype Actor language (Acore), and study of large scale experimental actor systems.

Approach

The MPSG is studying the architectural implications of large scale concurrent systems empirically, through the experimental implementation and analysis of sophisticated large scale concurrent systems, as well as theoretically, through development of the actor models of such systems. Each level of these systems is based on a fundamental, unifying approach: all processes are modeled as concurrent interactions between active message passing objects, actors.

To support the development and study of large scale actor systems, the MPSG is developing an operating environment for constructing and observing systems based on the Actor model of computation. This operating environment is built around Acore, a prototype core actor language used by the MPSG for constructing actor systems. Acore is an extensible, inherently concurrent language which combines the lexical scoping and higher order closures of functional languages with the encapsulation and message passing interfaces of object oriented languages. The result is an expression-oriented language for specifying complex actor behaviors. Other paradigms, such as rule-based or relational programming, are being integrated and unified with the functional and message passing paradigms of Acore to produce a coherent language illustrating the power of the Actor model.

The Actors operating environment includes the Acore compiler, debugging and analysis tools, and an actor machine emulator for running actor programs. The Acore compiler transforms each complex behavior into a system of simple actors whose messages can be concurrently processed by a low level concurrent architecture. The tools for observing and debugging Acore programs include Traveler, the actor observatory which displays the recorded history of events in a system of actors from different viewpoints, and the Dynamic Actor Analysis tool, which aids analysis of the concurrency and bottlenecks in actor programs.

The Actors operating environment will be used to construct, run, debug, and analyze the behavior of several large scale actor systems, from which the MPSG will derive implications for the architecture of general purpose large scale concurrent computers. This will also show that actors provide a highly appropriate methodology for developing concurrent systems, a methodology which spans generations of concurrent computers.

Progress

The MPSG has built working prototypes of the Actors operating environment subsystems. These prototypes implement useful subsets of their designed capabilities; for example, programs can be run and observed, but exception handling is minimal or nonexistent. These initial prototypes are being extended to fulfill their designed goals where possible; some, such as the Acore compiler, are being redesigned to provide more capability and flexibility.

Our research focuses on organizational structures occurring in these systems. Such structures can be modeled with Actors, which are universal primitives of concurrent computation. However, Actors need support for organizational operation, if they are to be practical for use in such large-scale Open Information Systems. The Organizations of Restricted Generality (ORGs) architecture is designed to meet this need. Each ORG has facilities for operations, reporting, membership, liaison, and management.

- The *Operations* facility provides resources (processors, storage, communications, etc.) for authorized tasks.
- The *Reporting* facility provides information for other facilities concerning what happened, when it happened, and which participants were involved.
- The *Membership* facility keeps track of the ORG population, creates new memberships, and terminates existing memberships as the ORG evolves.
- The *Liaison* facility controls communications crossing the ORG boundary. *Reception* is a subfacility of Liaison, which fields communications sent to the ORG as a whole. These boundaries make distributed garbage collection and graceful system evolution possible.

- The *Management* facility is responsible for the behavior of the ORG. It sets policies and procedures for ORG behavior.

Teams of Engaged Activity and Management (TEAMs) are suborganizations of an ORG which do not have a liaison facility and therefore have more flexibility in communication but correspondingly less control over communication between the TEAM and the rest of the containing organization.

These facilities are designed to make ORGs the organizational building blocks from which large-scale Open Information Systems can be constructed and managed. ORGs are basic, self-reliant but interdependent units of organizational structure and functionality from which larger, more complex systems can be composed.

The development of Open Information Systems Architecture will have important benefits. OIS Architecture can provide significant advantages in scalability and robustness, through the use of methods and principles similar to the ones used by human organizations. These methods and principles enable human organizations to work on large-scale tasks in a robust and productive fashion. They can provide the same benefits for human/telecomputer systems.

In addition, OIS Architecture can provide a better interface between humans and telecomputers; OIS Architecture works by principles familiar to humans, so people will understand the telecomputer systems more easily and intuitively. Such compatibility holds the promise of better interaction between humans and computers.

We have just started experimenting with the ORGs Architecture. Much work is left to do, in putting this architecture to use in the construction of Open Information Systems. For example, the Traveler system needs to be generalized for ORGs and TEAMs.

Server systems which schedule access to limited resources are important participants in Open Information Systems. We address several issues in the design and implementation of server systems that have *continuous availability*, so they may respond immediately to new messages, and for *graceful suspension and resumption* of service, to minimize the disruptive effects of downtime on the rest of the system. To illustrate these issues, this paper presents an implementation of a simple suspendable print server in the universal actor language InterAct, using *pipelining* and *complete behavior replacement* as modular implementation strategies. These facilities are designed to make ORGs the organizational building blocks from which large-scale Open Information Systems can be constructed and managed. ORGs are basic units of organizational structure and functionality from which larger, more complex systems can be composed. They provide support for both self-reliance and interdependence in Open Systems computing.

We have developed a preliminary implementation of ORGs, concentrating on Liaison first. Since Liaison implements the "boundaries" of ORGs, it is fundamental to the use of ORGs as modular components of organizations. Liaison also has implications for the Membership and Reporting facilities.

The *Organization Man(-ager)* is a tool which allows a graphic view of several ORGs, showing their members and the Liaison Actors that implement their external communication pathways. The graphic representation provides a convenient way of examining the contents of an ORG, manipulating Liaison Actors, and so forth. Further developments of this tool will focus on management of the organizational structure and behavior of ORGs, and towards monitoring the complex interactions of multiple ORGs.

The MPSG has developed theory to support the dynamic resolution of *Conflict* in ORGs. As is apparent from observation of human organizations, Conflict can occur at many different levels, and may be more or less extensive and/or tractable. An attempt to divide by zero is most likely a simple, localized, low-level Conflict, whereas a systems deadlock may represent high-level Conflict involving many ORGs. By developing an elemental "vocabulary" for describing Actor computations (e.g. features, constraints and guarantees about such computations), we hope to provide a broad language through which ORGs can support *Negotiation*. Negotiation involves the discovery, examination, resolution and/or avoidance of Conflict. Development of support for Negotiation is vital for the support of sophisticated, large-scale Open Information Systems. A "language" for describing problems and solutions, in organizational terms, will support both the development of Robust systems, as well as making human-computer interactions more tractable.

In addition, when OIS Architecture works by principles familiar to humans, people will understand the telecomputer systems more easily and intuitively. Conversely, modifying the structure and organization of OIS telecomputer systems should be quite natural when the systems themselves are specified and operated in an organizational paradigm. This human-telecomputer compatibility holds the promise of better interaction between humans and computers.

The ORG architecture is proving to be well suited to its function of supporting large-scale Open Information Systems. More research is still needed in several areas including security issues and the semantics of the membership facility.

Accomplishments

- Completed the integration of conceptual framework for robust, manageable Open Information Systems

- Began implementation of a more flexible Acore compiler, so the core can be shared for different source and target language implementations.

- Designed a true Logic Programming Language for Open Information Systems. In the context of Open Information Systems, it turns out that existing Horn clause languages do not exactly correspond with the capabilities of Deductive Inference. This extension to Acore integrates and unifies concurrent rule-based and relational paradigms with the message passing and functional paradigms already in Acore. It turns out that existing concurrent Horn clause languages make computational steps that are not Deductive Inference steps. Furthermore

- Improved syntax to make Acore more readable to programmers unfamiliar with Lisp-like languages and to make the language semantics simpler.
- Demonstrated how concurrent "Logic Programming" languages that rely on the use of concurrent shared "logic" variables for communication are inherently less efficient than languages based on message-passing.
- Demonstrated that Actors are as efficient as any other method for the implementation of concurrent languages based on Horn clauses.
- Completed the integration of conceptual framework for robust, manageable Open Information Systems
- Designed a more flexible Mobile C compiler, so the core can be shared for different source and target language implementations.

Principle Investigator Name:	Carl Hewitt
PI Institution:	MIT
PI Phone Number:	617-253-5873
PI E-mail address:	hewitt@ai.mit.edu
Grant or Contract Title:	The Actor Model
Grant or Contract Number:	N00014-88-K-0460
Reporting Period:	Final Report

2 List of Publications, Presentations, and Reports

Book

Carl Hewitt, Carl Manning, Jeff Inman and Gul Agha, editors, *Actors and ORGS for Open Systems Science*, MIT Press, Cambridge, Mass., forthcoming 1993.

Papers in Books not yet published

Carl E. Hewitt, "Towards Open Information Systems Semantics", in *Actors and ORGs for Open Systems Science*, Carl Hewitt, Jeff Inman, Carl Manning and Gul Agha editors, MIT Press, Cambridge, Mass., 1993

Carl E. Hewitt, "Logic Programming in Open Information Systems", in *Actors and ORGs for Open Systems Science*, Carl Hewitt, Jeff Inman, Carl Manning and Gul Agha editors, MIT Press, Cambridge, Mass., 1993

Carl E. Hewitt "Scientific Communities and Open Information Systems", in *Actors and ORGs for Open Systems Science*, Carl Hewitt, Jeff Inman, Carl Manning and Gul Agha editors, MIT Press, Cambridge, Mass., 1993

Carl R. Manning, and Carl Hewitt, "Meta-InterAct: An InterAct Meta-Interpreter", in *Actors and ORGs for Open Systems Science*, Carl Hewitt, Jeff Inman, Carl Manning and Gul Agha editors, MIT Press, Cambridge, Mass., 1993

Carl E. Hewitt, and Jeff Inman "Organization of Large-Scale Open Information Systems", in *Actors and ORGs for Open Systems Science*, Carl Hewitt, Jeff Inman, Carl Manning and Gul Agha editors, MIT Press, Cambridge, Mass., 1993

Journal Papers

Hewitt, C. E., "Offices are Open Systems," *ACM Transactions on Office Information Systems*, Vol. 4-3, July 1986, pp. 271–287. Reprinted in *The Ecology of Computation*, Huberman, B. A. (ed), North Holland, 1988.

Hewitt, C. E., "Metacritique of McDermott and the Logicists," *Computational Intelligence*, Vol. 3, No. 3, August 1987, pp. 185–189.

Hewitt, C. E., "Open Information Systems Semantics for Distributed Artificial Intelligence" *Journal of Artificial Intelligence* , Vol. 14, No. 1, January 1991. pp. 79–106.

Hewitt, C. E., "Open Information Systems Semantics for Distributed Artificial Intelligence" *Journal of Artificial Intelligence* , Vol. 14, No. 1, January 1991. pp. 79–106.

Hewitt, C. E. and Inman, J. T., "DAI Betwixt and Between: From 'Intelligent Agents' to Open Systems Science" *IEEE Transactions on Systems, Man, and Cybernetics* Special Issue on Distributed Artificial Intelligence, Vol. 21 No. 6. November-December 1991.

Refereed Conference Papers

Hewitt, C. E., and Agha, G., "Guarded Horn Clause Languages: Are They Deductive and Logical?" *Proceeding of the International Conference on Fifth Generation Computer Systems*, Nov. 28–Dec. 2, 1988, Tokyo, Japan.

Hewitt, C. E., "Can Office Technology Support Office Dialogues" *IFIP-89*, August 1989.

Hewitt, C. E., "Organizations of Restricted Generality." In *1989 IFIP Congress Proceedings*, IFIP, August 1989, San Francisco, CA.

Hewitt, C. E., "Towards Open Information Systems Semantics" *Proceedings of the 10th International Workshop on Distributed Artificial Intelligence*, October 23-27, 1990, Bandera, Texas.

Reports

Palmucci, J., Waldspurger, C., Duis, D., and Krause, P., "A Concurrent Interpreter for Guarded Horn Clauses," A.I. Memo No. 1106, MIT A.I. Laboratory, Cambridge, MA, August 1990.